BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images, please do not report the images to the Image Problem Mailbox.

WHAT IS CLAIMED IS:

1. An article of manufacture including program logic for performing configuration checking of a network, wherein the program logic causes operations to be performed, the operations comprising:

scanning a network data store for at least one transaction;

generating at least one event for said transaction;

associating at least one configuration policy with said event;

comparing said configuration policy with configuration data associated with said event; and

determining whether said configuration policy has been violated based on the comparison.

2. The article of manufacture of claim 1, wherein the operations further comprise:

generating at least one trigger for said event; and

associating said configuration policy with said trigger.

3. The article of manufacture of claim 1, wherein said configuration policy is retrieved from a local policy data store.

4. The article of manufacture of claim 3, wherein said configuration policy in the local policy data store is automatically updated with a configuration policy in a remote data store.

5. The article of manufacture of claim 1, wherein the operations further comprise:

receiving a hypothetical network scenario;

generating at least one transaction based on the hypothetical network scenario;

populating the network data store with configuration data for said transaction; and

after determining whether said configuration policy has been violated based on the comparison, rolling back said transaction.

6. The article of manufacture of claim 1, wherein the operations further comprise:

receiving a request to perform configuration checking on an existing network configuration.

7. The article of manufacture of claim 1, wherein the operations further comprise:

when said configuration policy has been violated, performing an action specified in that configuration policy.

8. The article of manufacture of claim 7, wherein the action is at least one of logging an indication that the configuration policy has been generated, generating at least one policy violation event, sending a notification, and highlighting a network topology viewer that graphically depicts the network.

9. The article of manufacture of claim 1, wherein the operations further comprise:

when said configuration policy has been violated,

accessing a solution in a knowledge data store; and

applying the solution so that said configuration policy is not violated.

10. The article of manufacture of claim 1, wherein the operations further comprise:

when said configuration policy has been violated,

determining that a component in the network is able to provide a solution; and

allowing the component to apply the solution so that said configuration policy is not violated.

11. The article of manufacture of claim 1, wherein the operations for determining whether said configuration policy has been violated further comprise at least one of identifying incompatibilities between components in the network, performance issues, and availability issues.

12. An article of manufacture including program logic for performing proactive configuration checking of a network, wherein the program logic causes operations to be performed, the operations comprising:

receiving a hypothetical network scenario;

generating at least one transaction based on said hypothetical network scenario;

populating a network data store with configuration data for said transaction;

generating at least one event for said transaction using a mapping of events to transactions; and

using configuration data associated with said event to determine whether a configuration policy has been violated.

13. The article of manufacture of claim 12, wherein the operations further comprise:

rolling back said transaction by removing the configuration data for said transaction from the network data store.

14. An article of manufacture including program logic for performing reactive configuration checking of a network, wherein the program logic causes operations to be performed, the operations comprising:

receiving a request to perform configuration checking on an existing network configuration;

scanning a network data store for at least one transaction;

generating at least one event for said transaction using a mapping of events to transactions; and

using configuration data associated with said event to determine whether a configuration policy has been violated.

15. The article of manufacture of claim 14, wherein the operations further comprise:

when said configuration policy has been violated, automatically correcting the violation.

16. An article of manufacture including program logic for correcting a configuration problem, wherein the program logic causes operations to be performed, the operations comprising:

discovering the configuration problem;

determining whether there is at least one solution for the configuration problem in a knowledge data store;

when it is determined that there is at least one solution in the knowledge data store, automatically selecting a solution to solve the configuration problem;

when said solution can be automatically applied, automatically applying said solution; and

when said solution cannot be automatically applied, notifying a user.

17. The article of manufacture of claim 16, wherein operations for detecting the configuration problem further comprise:

periodically interrogating said component in the network for data; and

determining whether there is a configuration problem based on the interrogation.

18. The article of manufacture of claim 16, wherein operations for detecting the configuration problem further comprise:

receiving at least one report from at least one component in the network; and

determining whether there is a configuration problem based on the report.

19. The article of manufacture of claim 16, wherein the operations further comprise:

when a component in the network identifies a solution, allowing the component to automatically apply the solution.

20. The article of manufacture of claim 16, wherein the configuration problem is at least one of a network configuration problem and a storage configuration problem.

21. A system for performing configuration checking of a network, comprising:

a processor;

a computer readable medium accessible to the processor; and

program logic including code capable of causing the processor to perform:

scanning a network data store for at least one transaction;

generating at least one event for said transaction;

associating at least one configuration policy with said event;

comparing the said configuration policy with configuration data associated with said event; and

determining whether said configuration policy has been violated based on the comparison.

22. The system of claim 21, wherein the code is capable of causing the processor to further perform:

generating at least one trigger for said event; and

associating said configuration policy with said trigger.

23. The system of claim 21, wherein said configuration policy is retrieved from a local policy data store.

24. The system of claim 23,wherein said configuration policy in the local policy data store is automatically updated with a configuration policy in a remote data store.

25. The system of claim 21, wherein the code is capable of causing the processor to further perform:

receiving a hypothetical network scenario;

generating at least one transaction based on the hypothetical network scenario;

populating the network data store with configuration data for said transaction; and

after determining whether said configuration policy has been violated based on the comparison, rolling back said transaction.

26. The system of claim 21, wherein the code is capable of causing the processor to further perform:

receiving a request to perform configuration checking on an existing network configuration.

27. The system of claim 21, wherein the code is capable of causing the processor to further perform:

when said configuration policy has been violated, performing an action specified in that configuration policy.

28. The system of claim 27, wherein the action is at least one of logging an indication that the configuration policy has been generated, generating at least one policy violation event, sending a notification, and highlighting a network topology viewer that graphically depicts the network.

29. The system of claim 21, wherein the code is capable of causing the processor to further perform:

when said configuration policy has been violated,

accessing a solution in a knowledge data store; and

applying the solution so that said configuration policy is not violated.

30. The system of claim 21, wherein the code is capable of causing the processor to further perform:

when said configuration policy has been violated,

determining that a component in the network is able to provide a solution; and

allowing the component to apply the solution so that said configuration policy is not violated.

31. The system of claim 21, wherein the code for determining whether said configuration policy has been violated is capable of causing the processor to further perform at least one of identifying incompatibilities between components in the network, performance issues, and availability issues.

32. A system for performing proactive configuration checking of a network, comprising:

a processor;

a computer readable medium accessible to the processor; and

program logic including code capable of causing the processor to perform:

receiving a hypothetical network scenario;

generating at least one transaction based on said hypothetical network scenario;

populating a network data store with configuration data for said transaction;

generating at least one event for said transaction using a mapping of events to transactions; and

using configuration data associated with said event to determine whether a configuration policy has been violated.

33. The system of claim 32, wherein the code is capable of causing the processor to further perform:

rolling back said transaction by removing the configuration data for said transaction from the network data store.

34. A system for performing reactive configuration checking of a network, comprising:

a processor;

a computer readable medium accessible to the processor; and

program logic including code capable of causing the processor to perform:

receiving a request to perform configuration checking on an existing network configuration;

scanning a network data store for at least one transaction;

generating at least one event for said transaction using a mapping of events to transactions; and

using configuration data associated with said event to determine whether a configuration policy has been violated.

35. The system of claim 34, wherein the code is capable of causing the processor to further perform:

when said configuration policy has been violated, automatically correcting the violation.

36. A system for correcting a configuration problem, comprising:

a processor;

a computer readable medium accessible to the processor; and

program logic including code capable of causing the processor to perform:

discovering the configuration problem;

determining whether there is at least one solution for the configuration problem in a knowledge data store;

when it is determined that there is at least one solution in the knowledge data store, automatically selecting a solution to solve the configuration problem;

when said solution can be automatically applied, automatically applying said solution; and

when said solution cannot be automatically applied, notifying a user.

37. The system of claim 36, wherein the code for detecting the configuration problem is capable of causing the processor to further perform:

periodically interrogating said component in the network for data; and

determining whether there is a configuration problem based on the interrogation.

38. The system of claim 36, wherein the code for detecting the configuration problem is capable of causing the processor to further perform:

receiving at least one report from at least one component in the network; and

determining whether there is a configuration problem based on the report.

39. The system of claim 36, wherein the code is capable of causing the processor to further perform:

when a component in the network identifies a solution, allowing the component to automatically apply the solution.

40. The system of claim 36, wherein the configuration problem is at least one of a network configuration problem and a storage configuration problem.

41. A method for performing configuration checking of a network, comprising:

scanning a network data store for at least one transaction;

generating at least one event for said transaction;

associating at least one configuration policy with said event;

comparing the said configuration policy with configuration data associated with said event; and

determining whether said configuration policy has been violated based on the comparison.

42. A method for performing proactive configuration checking of a network, comprising:

receiving a hypothetical network scenario;

generating at least one transaction based on said hypothetical network scenario;

populating a network data store with configuration data for said transaction;

generating at least one event for said transaction using a mapping of events to transactions; and

using configuration data associated with said event to determine whether a configuration policy has been violated.

43. A method for performing reactive configuration checking of a network, comprising:

receiving a request to perform configuration checking on an existing network configuration;

scanning a network data store for at least one transaction;

generating at least one event for said transaction using a mapping of events to transactions; and

using configuration data associated with said event to determine whether a configuration policy has been violated.

44. A method for correcting a configuration problem, comprising:

discovering the configuration problem;

determining whether there is at least one solution for the configuration problem in a knowledge data store;

when it is determined that there is at least one solution in the knowledge data store, automatically selecting a solution to solve the configuration problem;

when said solution can be automatically applied, automatically applying said solution; and

when said solution cannot be automatically applied, notifying a user.